सत्यमेव जयते

INTELLECTUAL PROPERTY **INDIA**

GOVERNMENT OF INDIA
MINISTRY OF COMMERCE & INDUSTRY,
PATENT OFFICE, DELHI BRANCH,
W - 5, WEST PATEL NAGAR,
NEW DELHI - 110 008.

I, the undersigned, being an officer duly authorized in accordance with the provision of the Patent Act, 1970 hereby certify that annexed hereto is the true copy of the ***Application, Complete Specification*** *and* ***Drawing Sheets*** *filed in connection with Application for Patent No.* ***993/Del/02*** *dated* ***27th September 2002.***

Witness my hand this 17th Day of September 2003.

(S.K. PANGASA)
Assistant Controller of Patents & Designs

FORM 1
THE PATENTS ACT, 1970
(39 of 1970)
APPLICATION FOR GRANT OF A PATENT
(See Sections 5(2), 7, 54 and 135)

27 SEP 2002

1. I/we.

STMicroelectronics Pvt. Ltd., an Indian company, of Plot No. 2 & 3, Sector 16A, Institutional Area, Noida – 201 3001, Uttar Pradesh, India.

2. hereby declare –

(a) that I am/we re in possession of an invention titled *"Improved Mapping Of Programmable Logic Devices."*

(b) that the ~~provisional~~/ complete specification relating to this invention is filed with this application

(c) that there is no lawful ground of objection to the grant of a patent to me/us.

3. further declare that the inventor(s) for the said inventions is/are

(i) SHARMA Sunil Kumar, an Indian citizen, of A312, Sector 19 Noida, U.P. 201 301, India.

DUPLICATE

4. I/we claim the priority from the application(s) filed in convection countries, particulars of which are as follows: **NA**

5. I/we state that the said invention is an improvement in or modification of the invention the particulars of which are as follows and of which I/we are the applicant/patentee: **NIL**

6. I/we state that the application is divided out of my/our application, the particulars of which are given below and pray that this application be deemed to have been filed on under section 16 of the Act. **NIL**

7. That I am/we are the assignee or legal representative of the true and first inventors.

8. That my/our address for service in India is as follows:

ANAND & ANAND, Advocates
B-41, Nizamuddin East
New Delhi – 110 013

Tel Nos.: (11) 4355078, 4355076, 4350360
Fax Nos.: (11) 4354243, 4352060

9- I/We the true and first inventors of this invention or the applicant(s) in the convention country declare that the applicant(s) herein is/are my/our assignee or legal representative.

a) Sunil Kumar Sharma an Indian National of *A312, Sector 19 NOIDA , UP-201 301*

Signature

Dated this day of 2002

10- that to the best of my/our knowledge, information and belief the fact and matters stated herein are correct and that there is no lawful ground of objection to grant of patent to me/us on this application.

11- Following are the attachment with the application

(a) Complete specification (3 copies)
(b) Abstract
(c) Formal drawings
(d) Power of Attorney
(e) Form 1 (in triplicate)
(f) Form 3 (in duplicate)
(g) Fee Rs. 5000/- In cash/cheque/bank draft bearing no. , date On Bank.

I/We request that a patent may be granted to me/us for the said invention.

Dated this day of 2002

Signature
STMicroelectronics Pvt. Limited

To

The Controller of Patents
The Patent Office, Delhi

02-IND-119

Form 2

0 0 5 3 DEL 2

THE PATENTS ACT, 1970

2 7 SEP 2002

COMPLETE SPECIFICATION

[See Section 10]

'IMPROVED MAPPING OF PROGRAMMABLE LOGIC DEVICES'

ORIGINAL

STMicroelectronics Pvt. Ltd., *Plot No. 2 & 3, Sector 16A, Institutional Area, Noida – 201 301, Uttar Pradesh, India, an Indian Company*

The following specification particularly describes and ascertains the nature of this invention and the manner in which it is to be performed.

IMPROVED MAPPING OF PROGRAMMABLE LOGIC DEVICES

Field of the Invention

This invention relates to an improved method and system for mapping an electronic circuit to a programmable logic device (PLD). In particular, the invention relates to the utilization of cascade logic elements while mapping.

Background of the Invention

Programmable Logic Devices (PLDs) provide the capability of implementing a wide range of electronic circuits using the same physical device. This capability is exploited by configuring the device appropriately for each desired application. The configuring process involves the mapping of the target circuit onto the available resources of the PLD. Programmable Gate Array (PGAs) and Field Programmable Gate Arrays (FPGAs) are the most widely used PLDs. The architecture of these devices incorporates Look Up Tables (LUTs) that are configured for desired functionality. The efficiency of mapping algorithms is critical to the effective utilization of PLDs and FPGAs. In LUT-based FPGAs the mapping is implemented on the LUTs.

The conventional LUT-based FPGA mapping algorithms can be divided into two classes. The algorithms in the first class emphasize the minimization of the number of LUTs in the solution. This class includes "Chortle" and "Chortle-crf" algorithms by Francis, based on tree decomposition and bin packing techniques. The algorithms in the second class emphasize the minimization of the delay of the solution. This class includes "Flowmap" by Cong and Y. Ding. Which use flow based techniques in mapping with node duplication to reduce the logical depth of the mapped netlist **[2].**

A DAG-map (Direct Analysis Graph) method for FPGA technology mapping for delay optimization has been proposed by J. Cong et.al. **[1]**. This method utilizes a graph based technology mapping algorithm "DAG-Map", for delay optimization in lookup-table based FPGA designs. The algorithm carries out technology mapping and delay optimization on the entire Boolean network. As a preprocessing step in "DAG-Map", a general algorithm transforms an arbitrary n-input network into a two-input network with a corresponding increase in the network depth; Finally, a graph matching based technique which performs area optimization without increasing the network delay is used as a post processing step for

"DAG-Map". This method does not however utilize the cascade elements available with each LUT in the FPGA.

Another optimal technology mapping algorithm for delay optimisation in Lookup -Table based FPGA Designs has been proposed by Jason Cong et. al. [2]. This method proposes a polynomial time technology-mapping algorithm, called "Flow-Map", that optimally solves the LUT-based FPGA technology-mapping problem for depth minimization for general Boolean networks. A key step in "Flow-Map" is the computation of a minimum height K-feasible cut in a network, by network flow computation. This algorithm does effectively minimize the number of LUTs by maximizing the volume of each cut and by several post processing operations but it does not utilize the cascade elements with the LUT to further reduce the size of the logic.

A method for On Area/Depth Trade-off in LUT-Based FPGA Technology mapping has been disclosed in reference [3] by Jason Cong and Yuzheng Ding. In this method the area and depth trade off in LUT based FPGA technology mapping is proposed by performing a number of depth relaxation operations to obtain a new network with bounded increase in depth and advantageous for subsequent re-mapping for area minimization. The resulting network is then re-mapped to obtain an area-minimized mapping solution. By gradually increasing the depth bound for each design a set of mapping solutions with smooth area and depth trade-off is achieved. For the area minimization step, an optimal algorithm for computing an area-minimum mapping solution without node duplication is developed. However this method also does not talk about the area minimization by utilizing the cascade elements with each LUT.

Another method proposed by Jason Cong and Yuzheng Ding [4] proposes an integrated approach to synthesis and mapping that extends the combinatorial limit set up by the depth-optimal "Flow Map" algorithm. The new algorithm, "FlowSYN", uses global combinatorial optimisation techniques to guide the Boolean synthesis process during depth minimization. The combinatorial optimisation is achieved by computing a series of minimum cuts of fixed heights in a network based on fast network flow computation, and the Boolean optimisation is achieved by efficient OBDD-based implementation of functional decomposition. This method also does not utilize cascade elements.

The Object and Summary of the Invention

The object of this invention is to provide an algorithm for efficiently synthesizing electronic circuits by utilizing cascade elements in LUT-based FPGAs.

Another object of the invention is to provide a method and mechanism for maximum utilization of on-chip resources in LUT-based FPGAs hence reducing the area in logic device.

Yet another object of the invention is to provide a method for synthesizing a logic circuit with minimum depth in LUT-based FPGAs.

It is yet another object of the invention to provide a method for realizing faster LUT-based FPGA implementation.

To achieve these and other objectives this invention provides an improved method for mapping an electronic digital circuit to a Look Up table (LUT) based Programmable Logic Device (PLD). The method operates by selecting an unmapped or partially mapped LUT, and identifying a group of circuit elements for mapping on the selected LUT based on the available capacity of the selected LUT and the mapping constraints. The identified circuit elements are then mapping onto the selected LUT. The identification of circuit elements and mapping is carried out while taken into consideration the Cascade Logic associated with the selected LUT. The process is continued until all the circuit elements have been mapped. The group of circuit elements is mapped to the cascade logic prior to mapping on the LUTs. Conversely, the cascade logic is incorporated only after either all circuit elements have initially been mapped onto LUTs or some circuit elements remain unmapped even after all LUTs have been utilized. The mapping constraints include timing constraints, placement constraints, and size constraints.

Brief Description of the Drawings

Figure 1 shows a schematic diagram of a conventional Field Programmable Gate Array. (FPGA)

Figure 2 shows the FPGA circuit implementation process, according to this invention.

Figure 3 shows the flow diagram of the technology-mapping step in the FPGA circuit implementation process, according to this invention.

Figure 4 shows the process flow diagram for LUT mapping in the technology-mapping step in the FPGA circuit implementation process, according to this invention.

Figure 5 shows the schematic diagram of a cascade element coupled with the LUT.

Figure 6 shows the schematic diagram of a typical four input logic cone.

Figure 7 shows a conventional net list mapped logic circuit without using the cascade feature of a programmable logic device.

Figure 8 shows a netlist mapping using the cascade feature of the programmable logic device in accordance with the present invention.

Detailed Description

The following description is based on the definition of technical terms given below:

Electronic design the logical structure of an electronic device such as an integrated circuit. This may be specified either as a behavioural description, as high-level Boolean equations, a circuit schematic or in any other form representing the logical arrangements of a device. It may include different constraints such as timing constraints, placement constraints, or mapping constraints etc.

Target hardware device the hardware device on which an electronic design is implemented. In context of this invention, a target hardware device typically includes the symmetric array of uncommitted logic elements. The uncommitted elements consist of the LUT, cascade gate, MUXes and flip-flops. These elements are grouped to form another hierarchy called programmable logic blocks, which are again grouped to form a programmable logic device.

Compiler software and hardware on which the software operates for compiling the electronics design. The function of the compiler is to synthesize the netlist and map the netlist (design) to the target device.

Mapping refers to the process of grouping gates from a gate netlist or other hardware independent representation of logic into a logic block. In other words the logic design may be divided into clusters representing the various logic functions within the design. These clusters are mapped onto the uncommitted logic elements in the programmable logic device during the compilation of the electronic design. The conditions for grouping (mapping) the gates into logic cells is that it should be possible to implement the grouped gates in the logic cell, For example, if a four input LUT is taken as a logic cell then the grouped gates must have less than or equal to four inputs for successful mapping.

The present invention utilizes the cascade elements during the HDL compilation and maps the design for programmable integrated circuits during technology mapping.

In the accompanying diagrams (**figure 6, figure 7, figure 8)** a hexagon represents a LUT, a circle represents a logic node, an arrow represents a connection, and a rectangle represents a cascade element.

Figure 1 is a schematic block diagram of a typical LUT-based Field Programmable Gate Array. A typical FPGA has vertical/horizontal routing lines **1**, an array of logic blocks **2** and interfacing I/O pads **3**. The routing resources **1** connect the logic block elements **2** and I/O pads. The FPGA can also have switch boxes at the intersection of routing lines for connecting to the logic block arrays.

Figure 2 shows the flow chart **100** of an electronic design compilation process. The process of compilation is started **110** by either clicking icons or by passing the command to start the

compilation. Once the compilation is started the design entries **120** are entered by the user according to which the circuit is synthesized **130.** The LUTs and the PLBs are then mapped **140** over the synthesized circuit. After proper placement of the elements and routing lines **150** the configuration bit is generated and the logic is then configured on the FPGA **170.** The compiler operates in accordance with user specifications. The user specifications can be in terms of timing requirements, area constraints or any other desired constraints. The compiler synthesizes the design to produce a net list, which describes the functionality of design for implementation on the programmable logic device. The net list can be the collection of gates, state machines & macros etc. The nodes of net list are connected via nets and each net has a signal associated with it. The net list is synthesized to remove the redundant logic, and to meet specified constraints.

Figure 3 shows a simplified block representation **200** of e technology mapping step of the electronic design compilation. The technology mapping can broadly be divided in two parts, LUT mapping **140** and PLB packing **142.**

Figure 4 shows a flow chart of the LUT mapping process 300 according to the present invention. In this process the first step is to decompose the GATE netlist 301 into simple gates. The second step establishes the feasibility of pushing additional logic without violating the specified constraints including the fan-out constraints 330 in an LUT 340. If it is feasible to incorporate more logic onto the LUT then the logic is mapped accordingly 350, and the control are returns to block 330. If it is not feasible to incorporate any additional logic onto the LUT then the feasibility of incorporating the cascade elements is established. If this is possible then appropriate logic elements are mapped onto the cascade elements. If the possibility does not exist then another LUT is selected and the control are returns to block 330. The logic clusters are formed by grouping nodes. Whenever the cluster is k-infeasible, no more nodes are added to the cluster. Before starting a fresh cluster, the feasibility of incorporating the cascade element in the device architecture is determined. If the feasibility is established the next node is mapped to the cascade element. This decreases the block count for the logic. As the cascade elements are hard wired they do not use the routing resources and the resultant delay is less than that of a LUT.

Figure 5 shows the structure of a Programmable Logic Block (PLB) **400** in a LUT-based FPGA. LUT **410** having inputs **401,** and cascade logic gate **420,** is connected to multiplexer

430 having flip-flop **440** at its output. One of the inputs **413** to the cascade logic is from the cascade out of the previous stage and the output **431** of the cascade gate is the cascade in for the next stage 422.

Figure 6 show the schematic diagram of a typical four input logic cone **600**. The nodes **601, 602, 603, 604** are inputs from other logic cones or logic gates. Nodes **610, 620, 640** are logic gate, and the arrow represents the connection between the logic gates.

Figure 7 shows the schematic diagram **700** of a mapped netlist without using the cascade feature of the programmable logic device. Nodes **701, 702, 703, 704** etc. are the inputs from the other logic cones or logic gates, nodes **710, 720,** etc. are the logic gates, and the arrow represents the connection between the logic gates.

Figure 8 shows the flow chart for the mapping of a netlist **800** using the cascade logic of the programmable logic device according to the current invention. The mapping is implemented according to predefined criteria. The criteria may be logic density, speed etc. In each case forming of cones is started from nodes e.g. **810, 820, 860** etc. whose fanins are already mapped or from the nodes which are primary inputs e.g. **801, 802, 811** etc. The maximum possible number of nodes is mapped onto each available LUT to increase the density of logic. Each node required to be mapped to an LUT is selected according to the speed criteria. If it is not possible to map any more nodes to an LUT, the feasibility of mapping the node to the cascade element **890** of the LUT is established. If the feasibility exists then the density of the logic implemented is increased as more logic of the same logic block is utilized At the same time the speed of the implemented circuit also increases as the cascade elements **890** use hardwired connecting lines instead of the general routing resources of the FPGA.

Another method that can be applied for extracting the cascade chains is:

1. Prior to forming the LUTs the nodes (logic gates) to be mapped to the cascade elements are identified.

2. The entire netlist is then mapped to the LUTs without considering the cascade logic elements.

3. The identified nodes are then extracted from the mapped list as a post operation and mapped onto the cascade logic elements.

Since the cascade element is a single universal gate (NAND or NOR) certain constraints have be observed while mapping gates onto the cascade element. These constraints can be divided into two groups.

1. Conditions that are to be verified when starting a new chain.

2. Constraints to be verified while the cascade chain is being formed.

The first group of conditions that need to be verified when starting a new cascade chain are:

1. The number of common inputs to the fan in LUTs of the cascade element should be more than four.

2. The gate implemented in cascade element should not be of type XOR, XNOR or NOT.

3. Either one of the fan in gates should be in the Cascade Element or none should be in cascade element.

4. Either the gate implemented in the Cascade Element or its input LUTs should be multi fan out, but not both.

5. If the output of the cascade element is a primary output then the gate implemented inside the cascade element should not be of type 'AND' or 'NOR' gate.

The second set of conditions that are needed to be verified while the cascade chain is being formed are:

1. The gate implemented in cascade element should not be of type XOR, XNOR or NOT.

2. Either one of the fan in gates should be in Cascade Element or none should be in the Cascade Logic.

3. Only one of either the gate implemented in the Cascade Logic or its input LUTs should be multi fan out.

4. If the output of the cascade element is a primary output then the gate implemented in the cascade element should not be of type 'AND' or 'NOR'.

5. If the cascade element is multi fan out then there should not be more than one cascade element in the fan out list.

6. If the LUT is multi fan out then there should not be more than one cascade element in the fan out list.

This process of utilizing the cascade elements while mapping a logic circuit onto target architecture is the independent of the algorithm used for LUT synthesis. This method provides an optimal solution superior to existing methods without any extra traversal of the gate level netlist.

It will be apparent to those with ordinary skill in the art that the foregoing is merely illustrative intended to be exhaustive or limiting, having been presented by way of example only and that various modifications can be made within the scope of the above invention.

Accordingly, this invention is not to be considered limited to the specific examples chosen for purposes of disclosure, but rather to cover all changes and modifications, which do not constitute departures from the permissible scope of the present invention. The invention is therefore not limited by the description contained herein or by the drawings, but only by the claims.

References:

[1] Kuang-Chien Chen, Jason Cong, Yuzheng Ding, Andrew Kahng, Peter Trajmar; *DAG-Map: Graph Based FPGA Technology Mapping For Delay Optimization;* IEEE Design and test of computers, pp 7-20, sept.1992.

[2] Jason Cong and Yuzheng Ding; *An Optimal Technology Mapping Algorithm for Delay Optimization in Lookup -Table Based FPGA Designs;* IEEE Trans. On Computer Aided Design of Integrated Circuits and Systems CAD, Vol.13, pp 1-12 Jan 1994.

[3] Jason Cong and Yuzheng Ding; *On Area/Depth Trade-off in LUT-Based FPGA Technology Mapping;* 30th ACM/IEEE design Automation Conference (DAC), pp. 213-218, 1993.

[4] Jason Cong and Yuzheng Ding; *Beyond the Combinatorial Limit in Depth Minimization for LUT-Based FPGA Designs;* IEEE/ACM International Conference on Computer Aided Design (ICCAD), pp. 110-114, Nov. 1993.

We claim:

1. An improved method for mapping an electronic digital circuit to a Look Up table (LUT) based Programmable Logic Device (PLD) comprising the steps of:
   - selecting an unmapped or partially mapped LUT,
   - identifying a group of circuit elements for mapping based on the available capacity of the selected LUT and the mapping constraints,
   - mapping the group of circuit elements onto the selected LUT, and
   - continuing the process of selecting an LUT, forming a group of circuit elements and mapping until all the circuit elements have been mapped,

   characterized in that, the cascade logic associated with each LUT is also incorporated in the steps of forming the group of circuit elements and the mapping of the group.

2. An improved method as claimed in claim 1 wherein said group of circuit elements are mapped to the cascade logic prior to mapping on the LUTs.

3. An improved method as claimed in claim 1 wherein the cascade logic is incorporated only after either all circuit elements have initially been mapped onto LUTs or some circuit elements remain unmapped even after all LUTs have been utilized.

4. An improved method as claimed in claim 1 wherein the mapping constraints include timing constraints, placement constraints, and size constraints.

5. An improved method as claimed in claim 1 wherein the mapping on the Cascade logic incorporates one or more of the following constraints depending upon the connectivity of the architecture:
   - XOR, XNOR and NOT functions are not mapped on the cascade logic,
   - only one of either the gate mapped onto the cascade logic or its input LUTs have multiple fan-outs,
   - if the output of the cascade logic is a primary output, then the gate mapped onto it is not an 'AND' or 'NOR' gate,
   - if the mapped gate has multiple fan outs then the outputs are not connected to more than one other gate mapped into a cascade logic element, and

- if the mapped gate connects to the output of a multi-fan out LUT then the output of the LUT is not connected to more than one cascade logic element.

6. An improved method as claimed in claim 1 including the verification of one or more of the following conditions at the initial mapping of the cascade logic chain depending upon the connections of the architecture:
   - the number of common inputs to the fan-in LUTs of the cascade logic is not greater than the number of inputs of the LUT,
   - the gate mapped onto the cascade logic is not of the type XOR, XNOR or NOT, and
   - only one of either the gates mapped on top the cascade logic or its input LUTs is multi fan.

7. An improved system for mapping an electronic digital circuit to a Look up table (LUT) based Programmable Logic Device (PLD) comprising:
   - selecting means for selecting an unmapped or partially mapped LUT,
   - grouping means for clustering circuit elements for mapping based on the available capacity of the selected LUT and the mapping constraints,
   - mapping means for mapping the group of circuit elements onto the selected LUT, and

   characterized in that, the grouping means and mapping means include the mapping of cascade logic associated with the selected LUT.

8. An improved method for mapping an electronic digital circuit to a Look Up table (LUT) based Programmable Logic Device (PLD) substantially as herein described with reference to and as illustrated in figures 2 to 6&8 of the accompanying drawings.

9. An improved system for mapping an electronic digital circuit to a Look up table (LUT) based Programmable Logic Device (PLD) substantially as herein described with reference to and as illustrated in figures 2 to 6&8 of the accompanying drawings.

Dated this 27th day of September, 2002

Shanti Kumar

of **ANAND & ANAND, Advocates**
Agents for the Applicants

ABSTRACT

This invention provides an improved method for mapping an electronic digital circuit to a Look Up table (LUT) based Programmable Logic Device (PLD). The method operates by selecting an unmapped or partially mapped LUT, and identifying a group of circuit elements for mapping on the selected LUT based on the available capacity of the selected LUT and the mapping constraints. The identified circuit elements are then mapping onto the selected LUT. The identification of circuit elements and mapping is carried out while taken into consideration the Cascade Logic associated with the selected LUT. The process is continued until all the circuit elements have been mapped. The group of circuit elements is mapped to the cascade logic prior to mapping on the LUTs. Conversely, the cascade logic is incorporated only after either all circuit elements have initially been mapped onto LUTs or some circuit elements remain unmapped even after all LUTs have been utilized. The mapping constraints include timing constraints, placement constraints, and size constraints.

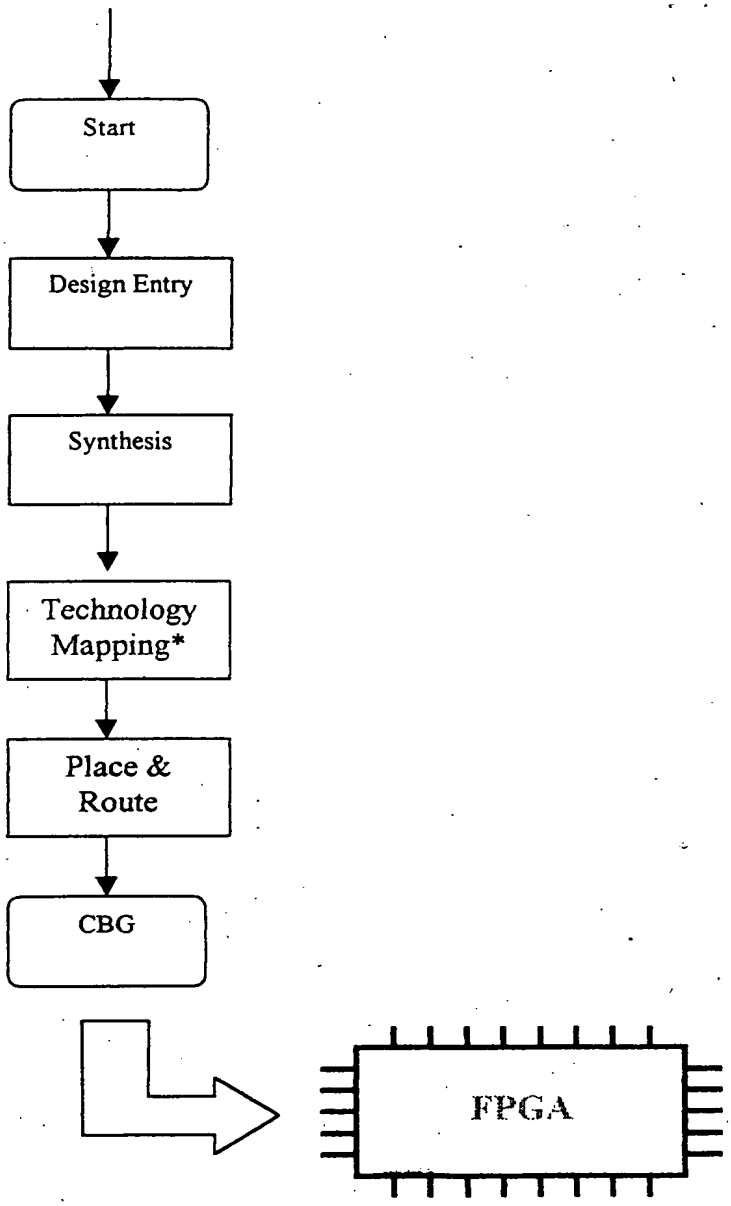


Fig. 1

Shanti Kumar

Of Anand And Anand Advocates
Attorney for the Applicant

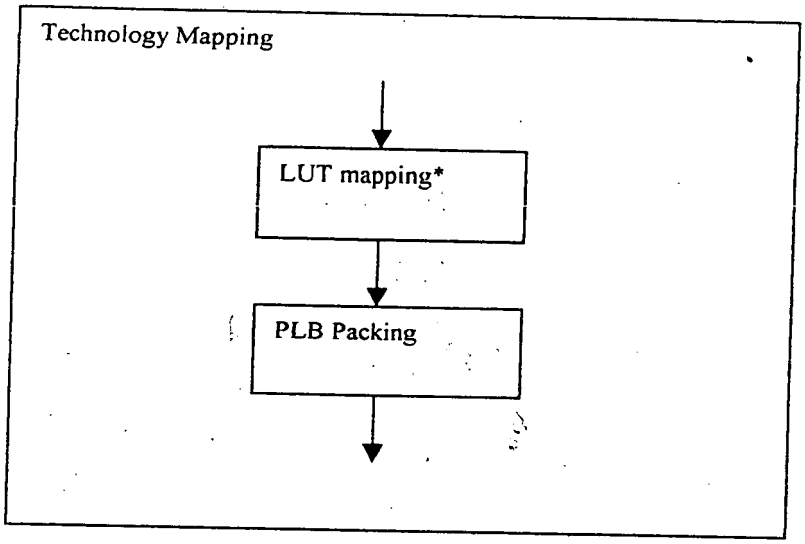


Fig. 2

Shanti Kumar

Of Anand And Anand Advocates
Attorney for the Applicant

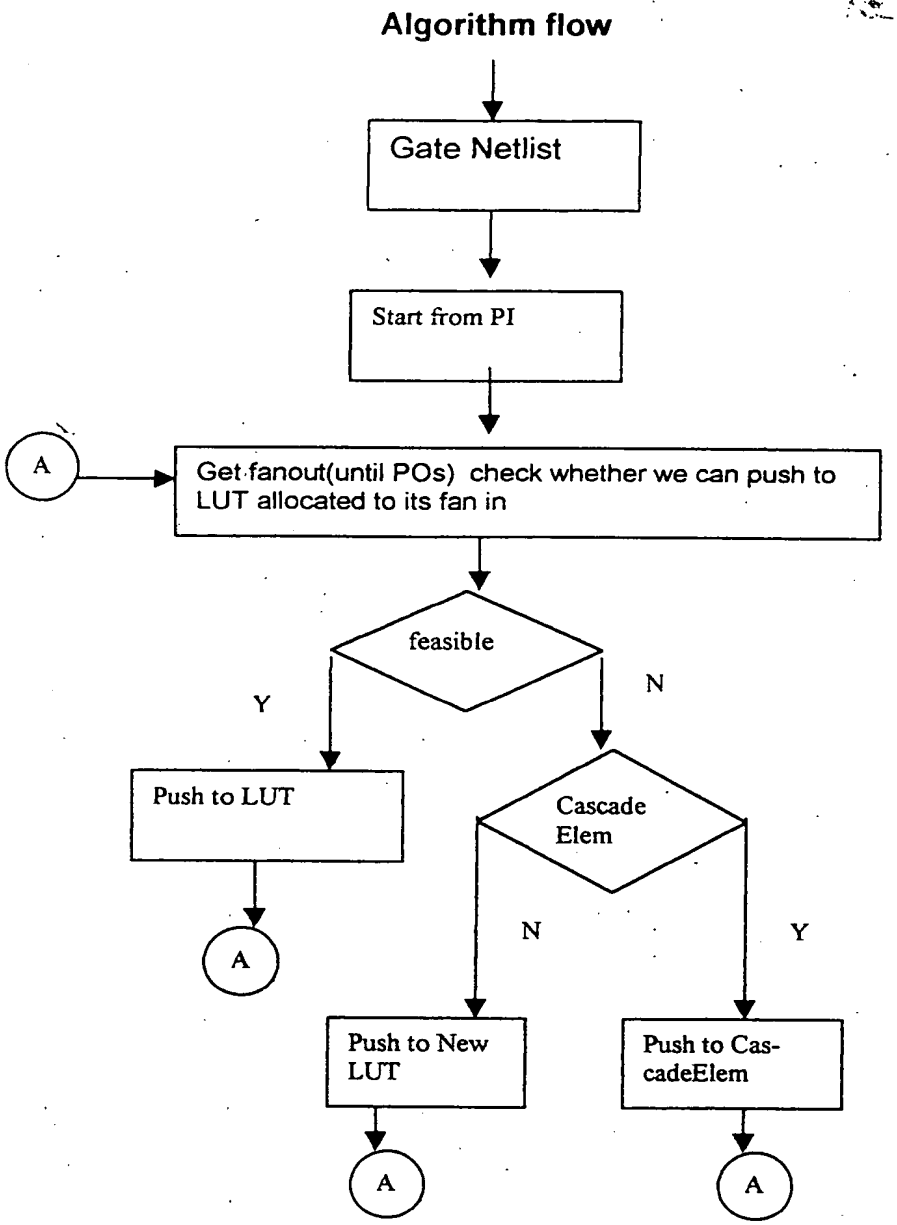


Fig. 3

Shanti Kumar

Of Anand And Anand Advocates
Attorney for the Applicant

Cascade Chain

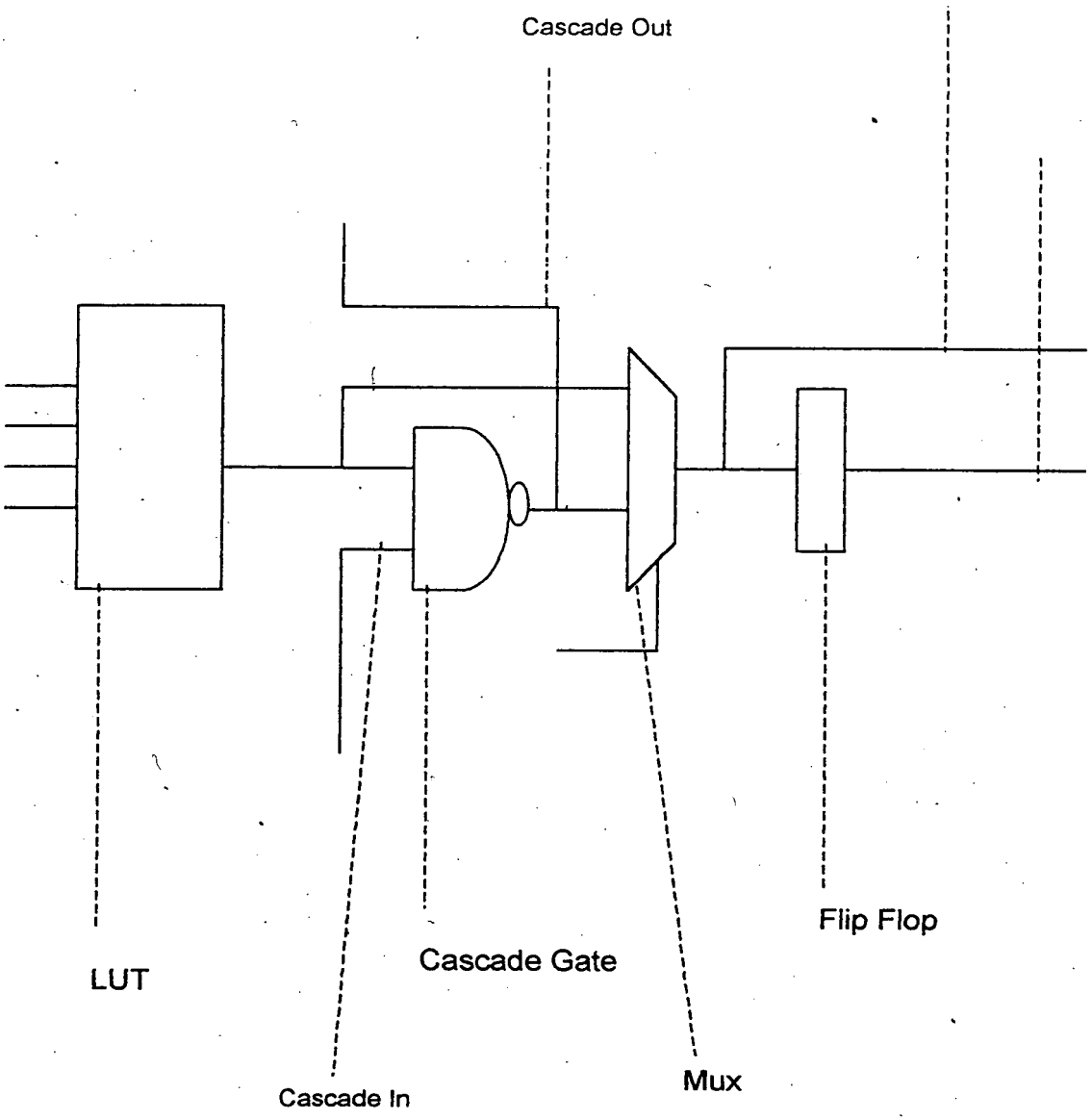


Fig. 4

Shanti Kumar

Of Anand And Anand Advocates
Attorney for the Applicant

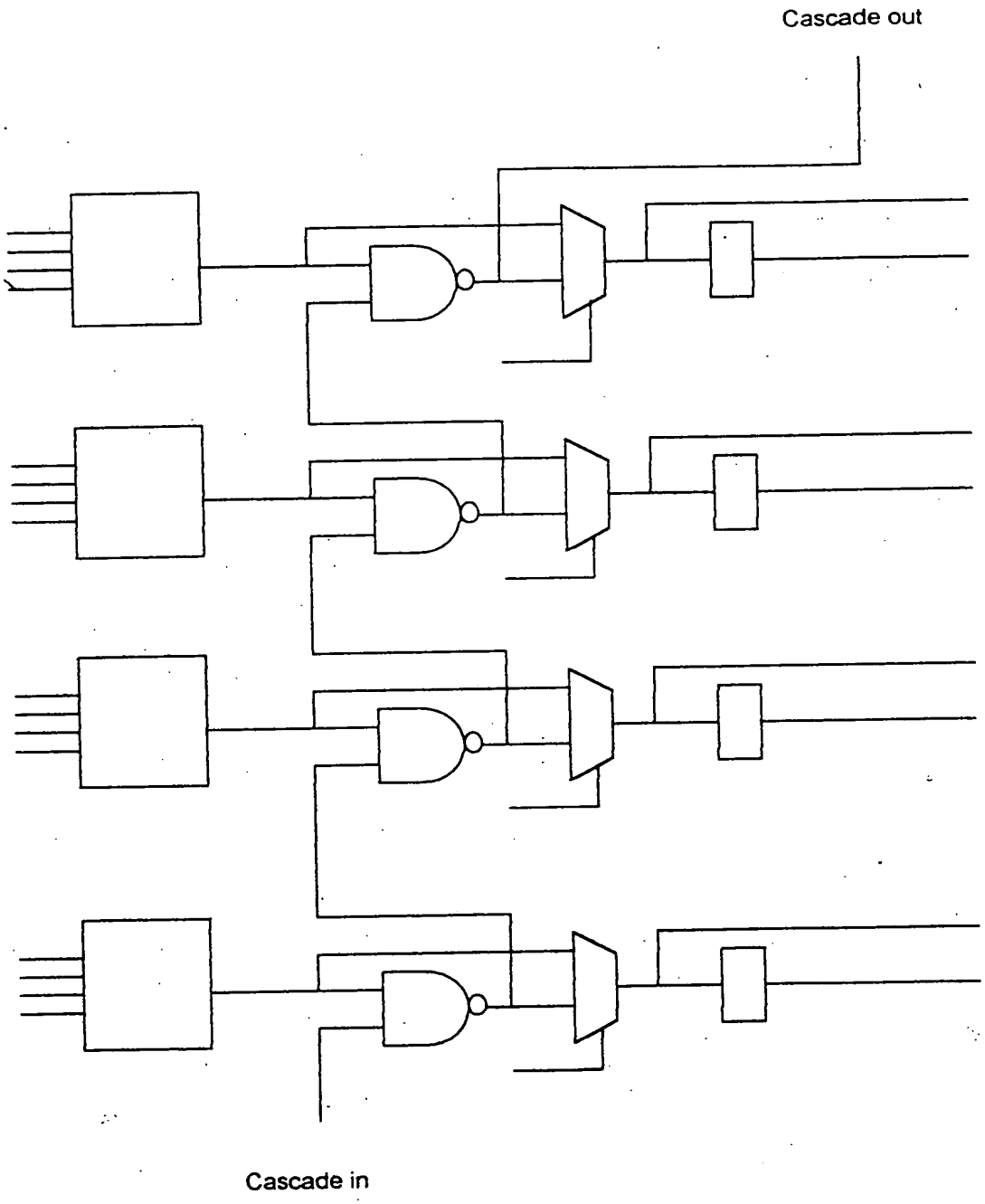


Fig. 5

Shanti Kumar

Of Anand And Anand Advocates
Attorney for the Applicant

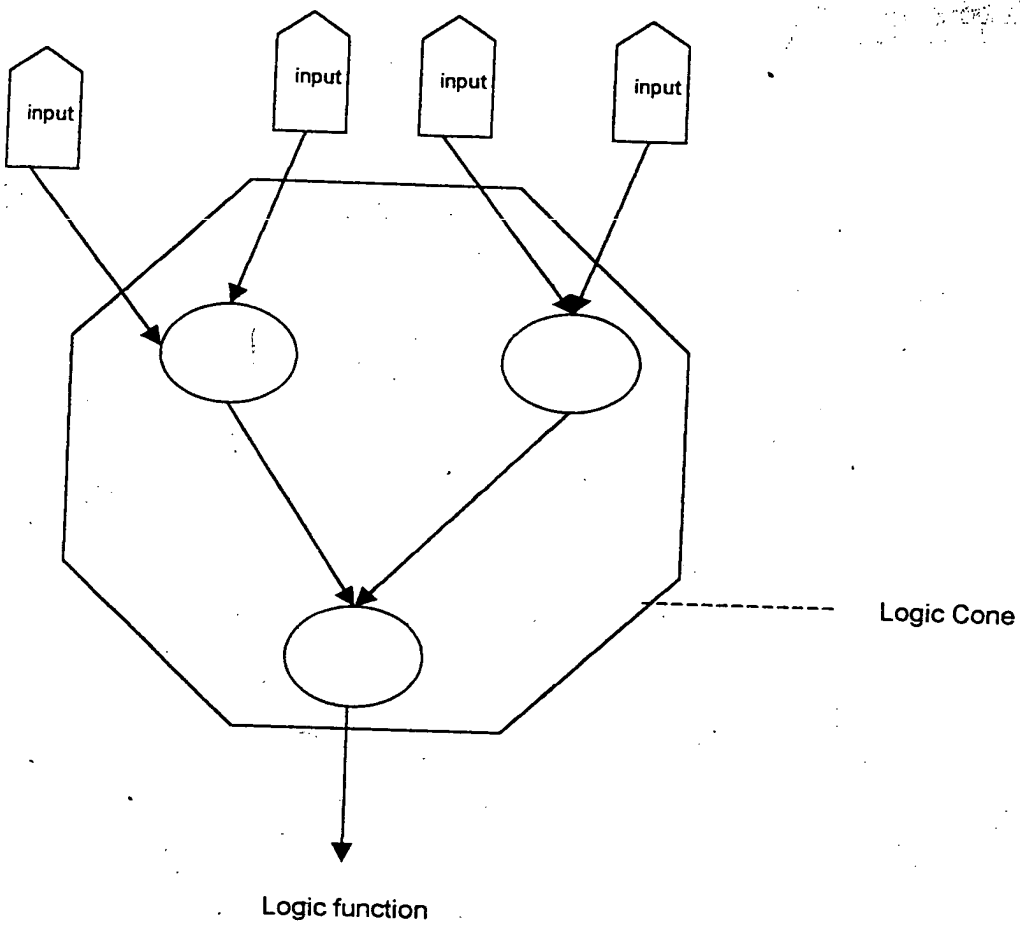
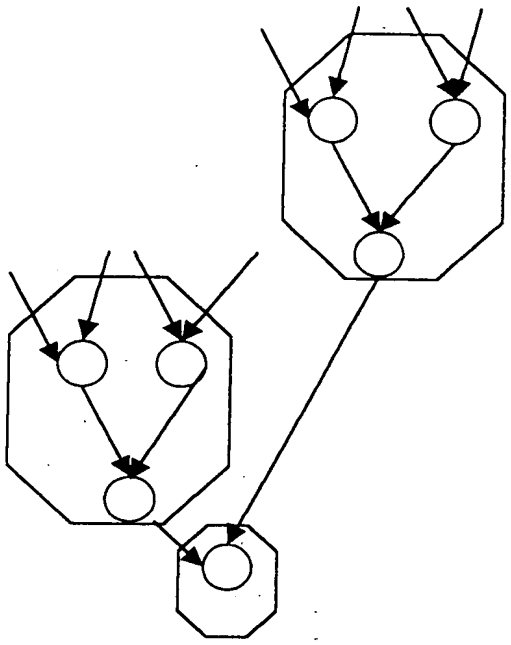


Fig. 6

Shanti Kumar

Of Anand And Anand Advocates
Attorney for the Applicant

Mapping for K=4
without cascade element



Depth: 2
Area: 3

Legend:

LUT

Cascade Element

Logic elements to be packed into LUTs / Cascade Elements.

Fig. 7

Shanti Kumar

Of Anand And Anand Advocates
Attorney for the Applicant

Mapping for K=4
with cascade element

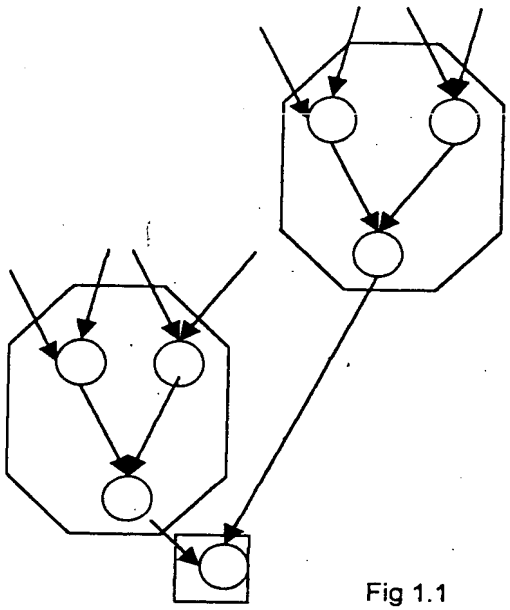


Fig 1.1

Depth: 1
Area: 2

Legend:

LUT

Cascade Element

Logic elements to be packed into LUTs / Cascade Elements.

Fig. 8

Shanti Kumar

Of Anand And Anand Advocates
Attorney for the Applicant